Unsupervised Video Tag Correction System

Hoang-Tung Tran, Elisa Fromont, François Jacquenet, Baptiste Jeudy, Adrien Martins

Unsupervised Video Tag Correction System

Hoang-Tung Tran*, Elisa Fromont*, Francois Jacquenet*,
Baptiste Jeudy*, Adrien Martins*

*Laboratoire Hubert Curien, UMR CNRS 5516
18 Rue du Professeur Benoît Lauras, 42000 Saint-Etienne
{hoang.tung.tran, elisa.fromont, francois.jacquenet}@univ-st-etienne.fr
{baptiste.jeudi, adrien.martins}@univ-st-etienne.fr

Abstract. We present a new system for video auto tagging which aims at correcting and completing the tags provided by users for videos uploaded on the Internet. Unlike most existing systems, we do not learn any tag classifiers or use the questionable textual information to compare our videos. We propose to compare directly the visual content of the videos described by different sets of features such as Bag-of-visual-Words or frequent patterns built from them. Then, we propagate tags between visually similar videos according to the frequency of these tags in a given video neighborhood. We also propose a controlled experimental set up to evaluate such a system. Experiments show that with suitable features, we are able to correct a reasonable amount of tags in Web videos.

1 Introduction

Classic text-based search engines already offer a good access to multimedia contents in the online world. However, they cannot index the extensive number of online videos unless these videos are carefully annotated before being put on the Web. However, user-provided annotations are often incorrect, i.e. irrelevant to the video (e.g. to increase the video's number of views), and incomplete. To overcome these drawbacks, we will focus on the task of setting up an automatic system to improve annotations of web videos. There have already been many efforts to automatically annotate videos (e.g (Morsillo et al., 2010), (Shen et al., 2011)). However, most of the proposed systems use limited concepts (tags) and some supervised information to learn one or many classifiers to tag a video dataset. These approaches thus seem inappropriate for any video on a large website such as Youtube where the number of possible tags is unlimited and where the true labels are inaccessible a priori. We thus would like to propose an unsupervised approach based on the comparison of the visual content of the videos to propagate the tags from the neighbor videos based on their textual frequency. In this approach the main scientific locks reside i) in the choice of the features that will be used to make relevant unsupervised comparisons, ii) in the comparison method itself, iii) in the propagation process and iv) in the evaluation of the entire system. A review of related works concerning the above mentioned problems is briefly given in Section 2. In Section 3, we describe in details how to apply data mining techniques as well as our proposed method to compare videos. The experiments done so far are presented in Section 4 and we conclude in Section 5.

2 General framework and related work

Finding relevant features (step 1 and 2). The first step of our process is to decompose a video into a sequence of keyframes (using for example (Zhuang et al., 1998)). Then, we describe the video based on the frames. Different features are usually best suited for different tasks. The current trend in computer vision is to concatenate different kinds of low level features in a high dimensional vector that will be subsequently used for solving the vision tasks. E.g, one can use edge distribution histograms, color moments or wavelet texture color autocorrelograms (Moxley et al., 2010), Histograms of Oriented Gradient (HOG) or audio features, LAB and HSV global color histograms, Haar or Gabor wavelets (see (Morsillo et al., 2010)). Another very popular technique is to construct a *Bag Of visual Words* (BoVW) from the original low-level feature vectors (see (Yang et al., 2007)). However, when using only the visual content to compare videos, the above-mentioned features might not be discriminative enough. Frequent pattern mining techniques are more and more often used in the computer vision community to get better features (see e.g. (Sivic and Zisserman, 2004), (Yuan et al., 2011) and, more recently, (Fernando et al., 2012)). Those approaches often rely on class information to be able to select a compact set of relevant features from the output of the mining algorithms.

Computing similarities between videos (step 3). Even though a video is considered as a sequence of images, variations in the videos duration or in the number of keyframes make them more difficult to compare. A first method consists in taking the average of all frames histograms (e.g. (Yang and Toderici, 2011)), to produce a single description for the whole video. The histogram can be thresholded to remove some potential noise. Here classical distance functions (e.g. $L1$) can be used to estimate the similarity between videos. Even if this method is efficient, one loses a lot of the available information by averaging all the frames. The second approach consists in comparing pairs of keyframes, e.g. computing the similarity between the two most similar frames of the videos as in Moxley et al. (2010). The comparison of the two videos is made using a unique pair of frames and no sequential information is taken into account. The last one makes use of common identical frames (but different in terms of formatting, viewpoints, camera parameters, etc.) called *near duplicate* to compare videos (see e.g. (Zhao et al., 2010)). These *near duplicate* can not be found in all the videos.

Tag propagation procedure (step 4). As most video auto tagging systems learn multiple classifiers, the tag propagation step is not needed. However, the *near duplicate*-based method presented in Zhao et al. (2010) use such propagation procedure on which ours is based. For each video V, a list of possible-relevant tags is obtained from the k most similar videos (using a K-nearest neighbor algorithm). After that, a score function is applied for each tag to estimate the relevance of that tag according to a given video V. This score function depends on the tag frequency, the number of tags associated to a video, and the video similarity. Finally, only the tags with a score greater than a threshold are considered suitable for the video V.

3 Improvement on the proposed auto tagging system

Proposed features As explained in Section 2, we can use many possible features to describe a video and this is a crucial point to work on to have a relevant tag propagation at the end of the process. We propose to use BOVW constructed from SIFT descriptors (Lowe, 2004) obtained regularly in each keyframe of a video as our low level features. We then want to use a pattern mining algorithm to extract better so called *mid-level* features to compare our videos. Most of the algorithms proposed in the literature take as input binary vectors. As explained by Fernando et al. (2012), the "binarization" of the original BOVW must be done carefully. We propose to use a simple equal-bin size discretization (with a number of bins equal to 4) for each visual word to transform our original histogram into a binary vector. Besides, the data mining techniques output a huge number of patterns (exponential in the number of dimensions of the binary vectors). Those patterns can be filtered out using supervised information as, e.g, shown in Fernando et al. (2012). However, in our case, no supervised information is available thus different criteria have to be proposed. We have thus decided to use the SLIM algorithm (Smets and Vreeken, 2012). This algorithm optimize a criterion based on the Minimum Description Length to reduce the number of output patterns to the ones that "well compress" the data. It employs a simple yet accurate heuristic to estimate the gain or cost of adding a candidate to the output pattern set. If F is the set of frequent patterns obtained using SLIM, we build a binary vector V of size $|F|$ for each keyframe. In this vector, $V(i)$ is set to 1 if the i^{th} pattern of F appears in this keyframe and 0 otherwise. Since the number of patterns in F can still be large, we also use a Principal Component Analysis (PCA) to reduce the dimension of vector V. Finally, the vector describing each keyframe is either only the BOVW histogram, only the vector V of SLIM patterns (reduced by PCA) or these two vectors concatenated.

Proposed asymmetrical video similarity measure The first step of our method consists in calculating all the pairwise similarities between all the keyframes of the videos. Then, we compute the average of all maximum similarities corresponding to one video. In other words, for each keyframe of a video A, we search in all the keyframes of video B for the highest pairwise matching score and we record this value. Then, we compute the average of all the recorded values for all the keyframes of the video A to return the similarity score of video A towards video B. If we denote $A(i)$ the i^{th} keyframe of A and $|A|$ the number of keyframes in A, then

$$sim(A, B) = 1/|A| \sum_i \max_j sim(A(i), B(j)).$$

The similarity $sim(A(i), B(j))$ between frames is just the inverse of a distance between the vectors representing the frames.

4 Experiments

We first performed a series of experiments on some image datasets to assess the interestingness of frequent patterns as features, the different distances and the PCA method on the output pattern histogram. Due to the lack of space, these experiments are not reported here but they showed that i) the frequent patterns (FP) can be interesting features compared to simple bag-of-words if they are carefully chosen; ii) the L1 distance can be a good distance measure to compare two high dimensional vectors describing a video (it is better than the usual intersection kernel used in computer vision to compare histograms); iii) a PCA where we keep enough components to explain 90% of the variance can help reducing the dimensionality of the feature vectors without damaging the accuracy.

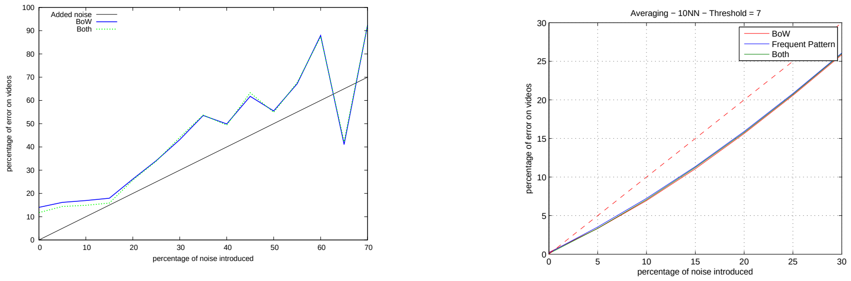

FIG. 1 – *Result of our tag correction algorithm on a real video dataset (left) and a synthetic one (right) using only bag of word features or bag of word and frequent patterns.*

The second series of experiments aim at proposing a new experimental protocol to evaluate the tag propagation method. We first use a 51 real videos dataset taken from a benchmark dataset of YouTube videos (Cao et al., 2009). Each video is decomposed into keyframes. There are about 27 keyframes for one video. The dimension of the SIFT-BOVW vocabulary is 1000. A video is thus represented by a matrix which contains for all keyframes of the video the visual word histogram which describes the frame. We kept this dataset reasonably small to be able to assess manually the interest of the original tags and the propagated ones for each video. The 51 videos were chosen such that they belong to 4 topics to ensure that this dataset contains pairs of similar videos and pairs of dissimilar videos. We then manually tag the videos with 35 tags. As the results on this dataset were not conclusive, we created a synthetic dataset of 182 videos built from 7 very different videos from the previous dataset. In both cases, were interested in evaluating the frequent pattern-based features compared to the BOVW-based features.

Tag propagation A video dataset is a triple (V, T, tag) where V is the set of videos $V = \{v_1, ..., v_n\}$, the set of possible tags is $T = \{t_1, ...t_m\}$ and tag is a relation on $V \times T$ such that $tag(v, t)$ is true if and only if video v has tag t. Our evaluation procedure is then:

- add some noise on the tags, i.e, choose a noise proportion $0 < p < 1$ and compute a noisy tag function tag_{noisy} such that, for each $t \in T$ and $v \in V$, with probability p we have: $tag_{noisy}(v, t) = \neg tag(v, t)$ (i.e. flip the value of a given tag with probability p);
- apply our tag correction technique, the output of the tag correction step is tag_{corr};
- compute the proportion of the incorrect tags after the correction step as: $err(tag, tag_{corr}) = \|\{(v, t) \in V \times T \mid tag(v, t) \neq tag_{corr}(v, t)\}\|/(\|V\|.\|T\|)$

The ideal case is $err(tag, tag_{corr}) = 0$. Notice that $err(tag, tag_{noisy}) \approx p$. This means that as soon as $err(tag, tag_{corr}) < p$, there is less incorrect tags on the noisy set after the tag propagation step than before. In Fig. 1, we plot the error $err(tag, tag_{corr})$ against the value of p. When the curve is below the diagonal line, we can state that our algorithm has decreased the number of incorrect tags.

Results on the real dataset We applied our evaluation procedure on the real 51 videos dataset presented at the beginning of this section. We averaged the results on 100 runs for each noise level. The results are presented in Fig. 1 (left). For almost all noise level, the number of incorrect tags is higher after our correction algorithm than before. These errors can be the result of the correction algorithm or the fact that the computed distance between videos does not reflect the real similarity of the videos. In particular, the number of videos that we use is quite small. In a dataset of millions of videos, the k nearest neighbors of a given video should be much more similar than in our small dataset (and thus have very similar tags). Another problem lies in the tags themselves: our algorithm use the visual similarity between videos to correct the tags. Thus it can be efficient only on tags that are correlated with the visual content.

Results on the synthetic dataset The maximum number of tags for this dataset is $182 * 7 = 1274$. This means that when adding 5% of noise in the dataset, 63 tag values are flipped in the dataset (some tags are added, some are removed). Then, to build the synthetic video we 1) choose randomly between 2 and 4 videos of the real video dataset; 2) choose randomly frames from each of the chosen real videos. The set of frames thus obtained is the synthetic video; 3) tag this synthetic video with A if it contains frames from video A, with B if it contains frames from video B and so on. Each synthetic video has therefore between 2 and 4 tags out of 7 possible tags. By this construction, if two synthetic videos share for instance the tag A, it means that they both contain similar frames extracted from the real video A. Moreover, by construction, each tag is associated with the visual content of the video. We therefore avoid the last problem encountered with the real dataset. For a noise level between 0 and 30%, we see on Fig. 1 (right) that the proportion of incorrect tags significantly decreases. For instance, at a noise level of 20%, the error proportion after tag correction is around 16%. The algorithm has thus removed about one quarter of the errors introduced by the noise. Note that for a higher level of noise, the number of incorrect tags is too large to expect improving the results by tag propagation.

Analysis of the results Although giving very promising results on the tag propagation as shown in Fig. 1 (right), the last series of experiments on the video datasets questions the usefulness of our pairwise video comparison method and of the proposed high level frequent pattern features. Indeed, the results using the pairwise comparison introduced in Section 3 are similar to the ones obtained using a simple averaging of the frames although the later one is more efficient to compute. Fig. 1 also shows that the frequent patterns built using the SLIM algorithm do not improve the tag comparison compared to simple BOVW features. The combination of both feature vectors also gives similar results which shows that for videos, on the contrary as for images, the patterns computed by the SLIM algorithm do not seem to give additional information compared to the BOVW from which they are built.

5 Conclusion

We have presented a complete unsupervised auto-tagging system which corrects and completes original tags on videos. The system seems effective especially when the number of videos in the dataset is sufficiently high to have a relevant enough neighborhood for each video. However, the new proposed features and the pairwise video comparison procedure do not seem to improve our results compared to baseline methods. As future work, we thus propose to take into account the sequential information in the video to create better high level features and to take into account the spatial position of the features in the frames. We also plan to work on the scalability of the proposed system to tackle larger real datasets.

References

Cao, J., Y. Zhang, Y. Song, Z. Chen, X. Zhang, and J. Li (2009). Mcg-webv: A benchmark dataset for web video analysis. Technical report, ICT-MCG-09-001.

Fernando, B., E. Fromont, and T. Tuytelaars (2012). Effective use of frequent itemset mining for image classification. In *European Conference on Computer Vision*, pp. 214–227.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*(2), 91–110.

Morsillo, N., G. S. Mann, and C. Pal (2010). Youtube scale, large vocabulary video annotation. In *Video Search and Mining*, Volume 287 of *Studies in Computational Intelligence*, pp. 357–386. Springer.

Moxley, E., T. Mei, and B. Manjunath (2010). Video annotation through search and graph reinforcement mining. *IEEE Transactions on Multimedia 12*(3), 184–193.

Shen, J., M. Wang, S. Yan, and X.-S. Hua (2011). Multimedia tagging: past, present and future. In *Proceedings of the 19th ACM international conference on Multimedia*, pp. 639–640.

Sivic, J. and A. Zisserman (2004). Video data mining using configurations of viewpoint invariant regions. In *Computer Vision and Pattern Recognition (1)*, pp. 488–495.

Smets, K. and J. Vreeken (2012). Slim: Directly mining descriptive patterns. In *SIAM International Conference on Data Mining*, pp. 236–247.

Yang, J., Y. Jiang, A. Hauptmann, and C. Ngo (2007). Evaluating bag-of-visual-words representations in scene classification. In *International workshop on multimedia information retrieval*, pp. 197–206. ACM.

Yang, W. and G. Toderici (2011). Discriminative tag learning on youtube videos with latent sub-tags. In *Computer Vision and Pattern Recognition*, pp. 3217–3224.

Yuan, J., M. Yang, and Y. Wu (2011). Mining discriminative co-occurrence patterns for visual recognition. In *CVPR: Conf. on Computer Vision and Pattern Recognition*, pp. 2777–2784.

Zhao, W., X. Wu, and C. Ngo (2010). On the annotation of web videos by efficient near-duplicate search. *IEEE Transactions on Multimedia 12*(5), 448–461.

Zhuang, Y., Y. Rui, T. Huang, and S. Mehrotra (1998). Adaptive key frame extraction using unsupervised clustering. In *Int. Conf. on Image Processing(1)*, pp. 866–870.

Résumé

Nous proposons un nouveau système de marquage automatique de vidéos visant à corriger et compléter automatiquement les “tags” fournis par les utilisateurs lors de la mise en ligne d’une nouvelle vidéo sur internet. Au contraire des systèmes existants, nous décidons de ne pas utiliser l’information textuelle possiblement fausse fourni par les utilisateurs ni de techniques d’apprentissage supervisé pour baser nos décisions. Nous comparons directement le contenu visuel des vidéos en nous basant sur des attributs discriminants appris lors d’une étape de fouille de motifs fréquents. Ce papier décrit également une méthode simple de propagation des tags entre vidéos visuellement proches et un protocole expérimental permettant d’évaluer notre approche.